

PENERAPAN *OPEN STREET MAP* UNTUK MENCARI LOKASI ATM TERDEKAT DENGAN ALGORITMA *KRUSKAL* BERBASIS *SMARTPHONE ANDROID* (STUDI KASUS: LOKASI ATM DI KOTA BENGKULU)

Edwin Dwi Anggara Putra¹, Ernawati², Funny Farady Coastera³

¹²³Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu.
Jl.WR. Supratman Kandang Limun Bengkulu 38371A INDONESIA
(telp: 0736-341022; fax: 0736-341022)

¹edwin.anggara21@gmail.com,

²wier_na@yahoo.com,

³ffcoastera@gmail.com

Abstrak: Terbatasnya informasi tentang lokasi suatu mesin ATM menyebabkan kesulitan bagi seseorang saat hendak menggunakan fasilitas mesin ATM tersebut. Kesulitan akan semakin terasa ketika kebutuhan untuk menggunakan fasilitas mesin ATM tersebut bersifat mendesak. Pada penelitian ini akan dibuat sebuah aplikasi yang dapat menyediakan informasi tentang lokasi suatu mesin ATM terdekat berbasis android. Aplikasi ini menggunakan algoritma *Kruskal*. Aplikasi ini juga memanfaatkan fasilitas dari *OpenStreetMap* untuk petanya dan data titik koordinat lokasi ATM nya. Metode pengembangan sistem yang digunakan adalah model *Rapid Application Development* (RAD) dan perancangan sistem menggunakan *Unified Modelling Language* (UML). Hasil akhir dari sistem ini adalah lokasi ATM terdekat dari posisi pengguna dan rute menuju lokasi ATM yang ditampilkan dalam peta. Dapat disimpulkan bahwa, aplikasi ini dapat melakukan pencarian lokasi ATM terdekat dengan menggunakan Algoritma *Kruskal* serta memanfaatkan peta *OpenStreetMap* dan dapat memberikan informasi mengenai lokasi ATM tersebut. Informasi ATM yang ditampilkan berupa nama BANK, alamat lokasi ATM dan nilai nominal uang.

Kata Kunci: Pencarian Lokasi, ATM, *Kruskal*, *OpenStreetMap*, Android

Abstract: The limitation of information about ATM location causing the difficulties for someone when wanted to use the ATM's facility. That difficulty will grow more when the needs for that ATM's facility is in urgent. This study builds a search application that can provide information on the location of the nearest ATM based on Android. This application uses an *Kruskal* algorithm. This application utilizes facilities of *OpenStreetMap* for the maps and coordinates data of its ATM. Development system method used is the *Rapid Application Development* (RAD) and system design using the *Unified Modeling Language* (UML). The end result of this system is the nearest ATM location from the user's position and route to the ATM location is presented in a map. It can be concluded that this application can perform a search nearest ATM location using *Kruskal* Algorithm and utilizes map of *OpenStreetMap*

and can provide information about the ATM location. ATM information is displayed in the form of BANK's name, address of ATM location and nominal value of money.

Keywords: Location Retrieval, ATM, Kruskal, OpenStreetMap, Android.

I. PENDAHULUAN

Fasilitas umum identik dengan pusat pelayanan masyarakat baik yang berkaitan dengan pemerintah, perekonomian, keamanan, maupun kebutuhan sehari-hari. Fasilitas umum ini tentunya menjadi sarana yang sangat penting dan dibutuhkan oleh masyarakat. Mobilitas masyarakat kota menjadi salah satu alasan mengapa diperlukannya sarana umum untuk membantu kelancaran dalam melakukan suatu kegiatan.

Salah satu fasilitas umum yang paling sering dicari oleh masyarakat adalah mesin ATM (Anjungan Tunai Mandiri). Mesin ATM adalah suatu alat elektronik yang dapat melayani nasabah dalam proses transaksi uang seperti penarikan uang tunai, cek saldo, dan transfer uang ke rekening orang lain. Dewasa ini mesin ATM juga bisa melakukan proses transaksi pembayaran, seperti pembayaran rekening listrik, telepon, asuransi, pajak, dan lain lain. Oleh karena itu, mesin ATM sudah menjadi kebutuhan umum masyarakat modern.

Di zaman sekarang ini *smartphone* Android mengalami perkembangan yang cepat didukung oleh kemampuan yang semakin baik serta harga yang semakin murah, maka *smartphone* dengan sistem operasi Android banyak dipilih masyarakat disemua kalangan dalam menyelesaikan kebutuhan sehari-hari. Setiap harinya banyak produsen yang mempromosikan *smartphone* Android miliknya dengan pilihan yang beragam, serta harga yang ditawarkan semakin murah dengan spesifikasi yang

lebih baik, maka tidak heran jika sekarang *smartphone* Android dapat dimiliki oleh semua kalangan mulai dari yang berekonomi tinggi sampai yang berekonomi rendah. Dengan demikian ponsel tidak lagi menjadi barang mewah, sehingga kebutuhan ponsel cerdas semakin tinggi, perputaran pelanggan ponsel di Indonesia diprediksi mencapai 8,6 persen dalam sebulan [1].

Seiring dengan perkembangan ilmu dan teknologi, muncul suatu perkembangan teknologi yaitu GPS (*Global Positioning System*). Bertujuan untuk mengetahui letak tempat yang akan dituju dan mengetahui keberadaan pengguna dengan bantuan sinyal satelit, GPS sendiri dapat memberikan informasi yang tepat dan akurat mengenai posisi, kecepatan, arah dan waktu. GPS ini sendiri sudah tertanam pada *smartphone* canggih dengan sistem operasi Android, dengan demikian setiap pengguna dapat mengetahui posisi pengguna tanpa takut tersesat disuatu tempat yang tidak diketahuinya. Keberadaan teknologi GPS pada Android ini dapat dimanfaatkan sebagai pemandu dalam pencarian suatu tempat, yaitu ATM (*Automated Teller Machine*/Anjungan Tunai Mandiri) yang terdapat di kota Bengkulu. Masalah yang membuat ide tersebut muncul yaitu sulitnya mencari letak ATM terdekat bagi masyarakat kota Bengkulu serta masyarakat yang berasal dari luar kota Bengkulu karena jumlah nya yang banyak dan terus bertambah.

Dengan demikian solusi untuk mengatasi masalah di atas adalah dengan merancang dan membangun aplikasi pencarian ATM terdekat berbasis Android. Dengan adanya *smartphone* Android yang dimiliki semua kalangan dengan mudah, maka dapat digunakan pula untuk mencari letak ATM terdekat untuk kebutuhan transaksi pengguna.

Hal tersebut yang menjadi latar belakang dalam melakukan penelitian yang disajikan dalam penelitian ini. Aplikasi ini berguna untuk mencari letak-letak ATM yang berada di kota Bengkulu. Pengguna aplikasi ini dapat mencari secara langsung letak ATM terdekat kapanpun dan dimanapun di kota Bengkulu, sehingga tidak memerlukan waktu yang lama dalam pencarian letak ATM.

Penelitian terkait dengan pencarian lokasi pernah dilakukan sebelumnya oleh (Esther Irawati S, 2012) yang mana melakukan penelitian tentang Pencarian Lokasi Fasilitas Umum Terdekat Dilengkapi Dengan Rute Kendaraan Umum Lyn. Aplikasi ini merupakan aplikasi yang menampilkan informasi mengenai lokasi fasilitas umum yang ada di Kota Surabaya. Aplikasi yang dibuat ini berbasis *mobile* dan menggunakan algoritma A* untuk penentuan lintasan pada menu kendaraan umum. Fasilitas umum yang ditampilkan adalah ATM, rumah sakit, apotik dan bank.

Penelitian tentang pencarian dengan *OpenStreetMap* juga pernah dilakukan oleh Yustian Mantjoro (2014) yang pernah melakukan penelitian tentang Penggunaan Teknologi *Openstreetmap* Untuk GIS Fasilitas Pelayanan Umum Berbasis Android (Studi Kasus Kota Palu). Aplikasi yang dibangun oleh Yustian ini berbasis *mobile* dan pemetaanya memanfaatkan *OpenStreetMap*. Pelayan umum yang ditampilkan mencakup informasi sarana pelayanan, stasiun tv/radio, mesjid, gereja, pura, kelenteng, tni, polisi, rumah sakit, bank, atm, rumah makan, kafe, KFC, kedai, sekolah, sekolah tinggi, universitas, taman, tamasya, villa, hotel, motel, kantor pemerintah, pasar, bandara, SPBU, taxi, terminal di Kota Palu.

Penelitian pencarian dengan menggunakan *Google Map* pernah dilakukan oleh (Richard R F S, 2013) yang melakukan penelitian tentang

Implementasi Sistem Informasi Geografis Daerah Pariwisata Kota Semarang Berbasis Android Dengan *Global Positioning System* (GPS). Aplikasi yang dibangun ini berbasiskan sistem operasi android dan memanfaatkan penggunaan GPS. Aplikasi ini juga dibangun dengan menggunakan *Google Maps* Api sebagai fungsi utama peta. Aplikasi ini memberikan informasi dalam bentuk peta yang dapat digunakan sebagai referensi bagi wisatawan yang berkunjung ke Semarang. Selain itu, informasi mengenai fasilitas umum, seperti ATM dan Bank, bandara, mesjid, gereja, rumah sakit, dan lainnya juga diberikan dalam aplikasi ini.

Berdasarkan permasalahan dan penelitian terkait yang telah dijelaskan sebelumnya, dalam Tugas Akhir ini akan dibuat suatu aplikasi pencarian lokasi ATM berbasis *smartphone* android dengan mengimplementasikan Algoritma Kruskal sebagai penentu lokasi ATM terdekat dari lokasi pengguna.

II. LANDASAN TEORI

A. Titik Koordinat

Menurut (Basofi, 2013) dalam referensi [2], titik koordinat adalah titik pertemuan antara kedua garis lintang dan garis bujur. Garis lintang sering disebut dengan *latitude* dan garis bujur sering disebut dengan *longitude*. Garis lintang adalah garis dari atas ke bawah (*vertical*) yang menghubungkan kutub utara dengan kutub selatan bumi, sedangkan garis bujur adalah garis mendatar (*horizontal*) yang sejajar dengan garis khatulistiwa. [3].

Penulisan titik koordinat memiliki dua cara penulisan, yaitu *Decimal Degree* (DD) dan *Degrees Minutes Seconds* (DMS). Berikut ini cara penulisan titik koordinat :

1. *Decimals Degrees* (DD)

Longitude (Bujur) : 122.2056608

Latitude (Lintang) : -4.2824

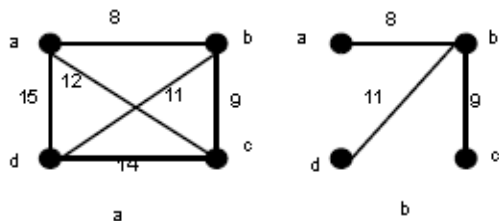
2. Degrees Minutes Seconds (DMS)

Longitude (Bujur) : 122° 12' 20.3796"

Latitude (Lintang) : -4° 16' 56.6394".

B. Minimum Spanning Tree (MST)

Pohon rentangan pada suatu graf adalah subgraf minimal yang menghubungkan semua simpul pada graf, apabila graf tersebut adalah graf berbobot (*Weighted Graph*), kemudian dari pohon rentang yang dimiliki oleh graf didefinisikan sebagai penjumlahan dari bobot – bobot seluruh cabang pada pohon rentang maka akan diperoleh pohon rentang yang memiliki bobot. Pohon rentang yang memiliki bobot terkecil pada suatu graph berbobot tersebut disebut Pohon rentang minimum (*Minimum Spanning Tree*) adalah metode yang digunakan untuk menentukan *spanning tree* yang menjangkau semua titik dengan total bobot sisi minimum pada graf terhubung dan berbobot. Contoh bentuk graf lengkap dan *Minimum Spanning Tree* yang dihasilkan pada Gambar 1 dibawah ini.



Gambar 1 (a) Graf berbobot dengan (b) Pohon rentangan minimumnya

Salah satu contoh aplikasi *Minimum Spanning Tree* secara langsung adalah permasalahan pemasangan jaringan dengan meminimasi jumlah penggunaan kabel/pipa untuk menghubungkan bangunan secara bersamaan (*connected*) atau pemasangan kabel jaringan telepon/listrik dengan menghitung minimalisasi biaya yang bisa

digunakan dan penggunaan kabel n sependek mungkin.

C. Algoritma Kruskal

Algoritma kruskal adalah sebuah algoritma dalam teori graf yang mencari sebuah *minimum spanning tree* untuk sebuah graf berbobot yang terhubung. Algoritma ini tergolong dalam Algoritma Greedy. Algoritma ini pertama kali muncul pada tahun 1956 dalam sebuah tulisan yang ditulis oleh Joseph Kruskal.

Pada Algoritma Kruskal, sisi (*edge*) dari graf diurut terlebih dahulu berdasarkan bobotnya dari kecil ke besar. Sisi yang dimasukkan ke dalam himpunan T adalah sisi graph G yang sedemikian sehingga T adalah *Tree* (pohon). Sisi dari Graph G ditambahkan ke T jika ia tidak membentuk *cycle*/sirkuit.

Secara Umum Algoritma Kruskal ditulis [4]:

1. T masih kosong
2. Pilih sisi/ruas (i,j) dengan bobot minimum
3. pilih sisi (i,j) dengan bobot minimum berikutnya yang tidak membentuk *cycle*/sirkuit di T, tambahkan (i,j) ke T
4. Ulangi langkah 3 sebanyak (n-2) kali.
5. Total langkah (n-1) kali

D. OpenStreetMap

OpenStreetMap API adalah suatu layanan *WEB* (*Web Map Service*) yang menyediakan akses langsung ke server basis data geospasial *openstreetmap*, tujuannya adalah hanya untuk mengakses basis data secara langsung. Hampir semua map editor untuk *openstreetmap* (OSM) menggunakan API ini. Karena server dari API didanai hanya melalui donasi maka penggunaan API untuk tujuan yang tidak jelas, seperti mengambil data dalam jumlah besar maka osm

akan memblok akses ke servernya tanpa suatu peringatan [5].

API ini sendiri menggunakan *style* layanan *web Representational State Transfer* (REST). Dalam layanan web REST, setiap data diberi dan beriteraksi melalui *Uniform Resource Identifier* (URI), yang lebih dikenal dengan alamat *web*. API ini juga dapat digunakan pada server lokal untuk tujuan mempercepat akses data. Terdapat beberapa operasi pengelolaan data dalam API seperti *create*, *read*, *update* dan *delete*. Untuk menggunakan operasi tersebut, diperlukan suatu autentikasi menggunakan akun *openstreetmap.org* berupa nama dan password atau melalui sistem autentikasi berbasis *web* seperti OAuth. API ini mempunyai sistem pendeteksi konflik untuk mencegah dua *mapper* mengubah fitur yang sama dalam waktu yang bersamaan. Versi API yang dapat digunakan yaitu API v0.6 yang disebarluaskan sejak 21 April 2009. API v0.6 ini merupakan komponen server yang mana permintaan REST dialamatkan. Permintaan REST menggunakan bentuk dari pesan HTTP *GET*, *PUT*, *POST* dan *DELETE*. Hasil dari permintaan tersebut adalah dalam bentuk XML, menggunakan MIME type "text/xml" dan encoding karakter UTF-8, dan boleh dikompresi dalam lapisan HTTP jika klien menyatakannya melalui HTTP "Accept" header yang dapat menangani pesan terkompresi. API dapat diakses melalui alamat web dari *openstreetmap* [6]:

Dalam permintaan *GET capabilities*, jika parameter yang diinputkan benar, sebuah metadata level layanan dalam bentuk dokumen XML akan dikembalikan, termasuk mengenai informasi lainnya yang tersedia di server. Untuk mengidentifikasi setiap perubahan maka digunakan suatu *tag changeset* dengan *attribute comment=**.

Penjelasan mengenai *attribute* pada XML hasil permintaan *GET* dapat dilihat pada tabel 11.

```
<osm version="0.6"
generator="OpenStreetMap server">
  <api>
    <version minimum="0.6"
maximum="0.6"/>
    <area maximum="0.25"/>
    <tracepoint per_page="5000"/>
    <waynodes maximum="2000"/>
    <changesets
maximum_element="50000"/>
    <timeout second="300"/>
  </api>
</osm>
```

Tabel 11 Atribut dokumen XML balasan

Atribut	Penjelasan
Version minimum dan maksimum	Versi API <i>call</i> yang disetujui oleh server.
Area maximum	Daerah maksimum dalam derajat persegi yang dapat diquery oleh API <i>call</i> .
Tracepoint per_page	Jumlah maksimum titik dalam sekali GPS <i>trace</i> .
Waypoint	Jumlah maksimum titik yang dapat diisi pada suatu jalan.
Changesets maksimum_elements	Jumlah maksimum kombinasi titik, jalan, relasi yang ada diisikan dalam <i>changeset</i> .

Untuk mendapatkan element suatu peta maka digunakan *bounding box* pada perintah **GET /api/0.6/map**. Penjelasan mengenai parameter permintaan data peta lihat pada Tabel 2

GET /api/0.6/map?bbox=left,bottom,right,top

Tabel 2 Parameter *Bounding Box*

Atribut	Penjelasan
Left	Longitude bagian kiri dari batas suatu kotak
Bottom	Latitude bagian bawah dari batas suatu kotak
Rights	Longitude bagian kanan dari batas suatu kotak
Top	Latitude bagian atas dari batas suatu kotak

Untuk permintaan akses suatu *changeset* dapat digunakan perintah:

GET /api/0.6/changeset

Berikut adalah *file format* dari *changeset* yang akan dikirimkan saat melakukan permintaan akses data keserver.

```

<osmChange version="0.3"
generator="Osmosis">
<modify version="0.3"
generator="Osmosis">
  <node id="12050350"
timestamp="2007-01-02T00:00:00.0+11:00"
lat="-33.9133118622908"
lon="151.117335519304">
    <tag k="created_by" v="JOSM"/>
  </node>
</modify>
</osmChange>

```

III. METODOLOGI

A. Teknik Pengumpulan Data

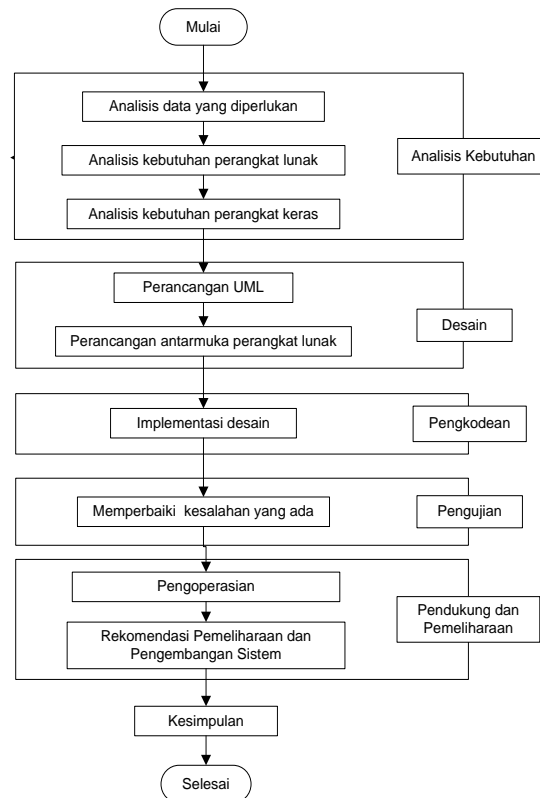
Langkah-langkah yang dilakukan dalam penelitian ini adalah:

1) *Studi Pustaka* : Studi kepustakaan dilakukan dengan mengumpulkan data dan informasi yang digunakan sebagai acuan dalam pembuatan aplikasi pencarian lokasi ATM Di Kota Bengkulu. Data dan informasi dapat berupa buku-buku ilmiah, laporan penelitian, skripsi, jurnal dan sumber-sumber tertulis lainnya yang berhubungan dengan pemahaman metode yang digunakan (Algoritma Kruskal), desain *Unified Modelling Language* (UML), pembuatan aplikasi dengan Eclipse, dan penggunaan *OpenStreetmap*.

2) *Observasi* : *Observasi* yang dilakukan dalam penelitian ini adalah dengan survei objek pengamatan dengan cara melakukan *tracking*/perjalanan dengan alat GPS untuk mendapatkan data titik koordinat ATM yang ada di Kota Bengkulu.

B. Metode Pengembangan Sistem

Tahapan dari model *waterfall* ini adalah mengarahkan kegiatan pengembangan dasar dari fase pertama hingga fase terakhir, alur metode sistem dapat dilihat pada gambar 2.



Gambar 2.. Diagram Alir Penelitian

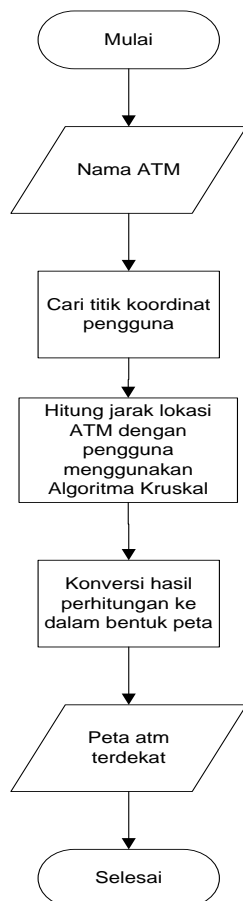
IV. ANALISIS DATA DAN PERANCANGAN

Teknologi yang berkembang pesat saat ini semakin mendukung terciptanya aplikasi atau sistem yang dibuat dengan tujuan untuk mempermudah aktifitas dan menyelesaikan permasalahan yang ada. Aplikasi yang dapat memberikan informasi mengenai keberadaan suatu lokasi sangat dibutuhkan terutama informasi mengenai lokasi yang sifatnya dibutuhkan dengan segera. Aplikasi harus dapat menampilkan informasi keberadaan lokasi terdekat dengan pengguna sistem. Salah satu contoh pencarian lokasi terdekat adalah kasus pencarian lokasi atm. Informasi mengenai lokasi atm terdekat sangat diperlukan oleh masyarakat pada masa sekarang, karena mesin atm sekarang tidak hanya berfungsi sebagai alat pengambil dan penyimpan uang saja, mesin atm sekarang juga bisa digunakan untuk melakukan proses pembayaran berbagai tagihan, seperti tagihan listrik, telepon dan sebagainya.

Pencarian lokasi terdekat dapat menggunakan algoritma Kruskal. Algoritma ini merupakan algoritma dalam teori graf yang mencari sebuah *minimum spanning tree* untuk sebuah graf berbobot yang terhubung.

A. Analisis Cara Kerja Sistem

Penerapan algoritma *Kruskal* pada aplikasi pencarian lokasi ATM terdekat di Kota Bengkulu berbasis android ini melakukan pencarian lokasi ATM terdekat dari keberadaan posisi pengguna. Hasil yang didapatkan berupa lokasi ATM dengan jarak terpendek dari posisi pengguna saat itu. Pengguna juga dapat melihat informasi mengenai ATM terdekat tersebut. Diagram alur kerja sistem dapat dilihat pada Gambar 3.



Gambar 3 Diagram Alur Kerja Sistem

Untuk menjalankan aplikasi pencarian ATM terdekat menggunakan algoritma *Kruskal* ini data yang digunakan adalah koordinat pengguna dan

koordinat lokasi ATM. Sebelum memulai pencarian, pengguna terlebih dahulu menginputkan nama ATM apa yang akan dicari. Setelah nama ATM dan titik koordinat pengguna didapatkan tahap selanjutnya adalah memulai melakukan perhitungan jarak tiap ATM dengan posisi pengguna berada. Tahap selanjutnya adalah mengkonversi hasil perhitungan tersebut ke dalam peta. Selanjutnya akan ditampilkan peta yang memuat lokasi ATM terdekat dari posisi pengguna berada.

B. Alur Metode Kruskal

Konsep dasar yang digunakan dalam algoritma Kruskal adalah pada setiap langkah, pilih sisi dari graf G yang berbobot minimum, tetapi sisi tersebut tidak membentuk sirkuit. Secara keseluruhan alur algoritma *Kruskal* dapat dilihat pada Gambar 4

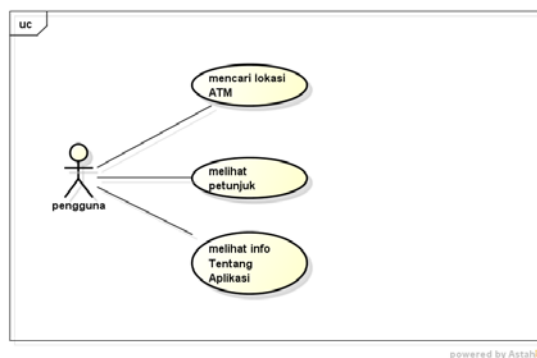


Gambar 4. Diagram Alur Algoritma Kruskal

C. Perancangan UML

Aplikasi ini dirancang dengan menggunakan pemodelan UML (*Unified Modeling Language*). UML dibutuhkan dalam pemodelan visual guna menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari setiap perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

1. *Use Case Diagram* : *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhasil menggunakan fungsi-fungsi itu. *Use case* diagram dari aplikasi yang akan dibangun dapat dilihat pada gambar 5.

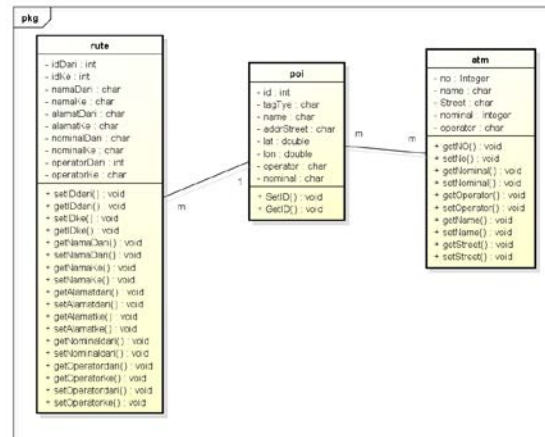


Gambar 5. *Use Case Diagram* Aplikasi

Dilihat dari gambar 4.3 pada sistem ini terdapat satu aktor yaitu pengguna. Aktor pengguna dapat mengakses menu Cari Lokasi ATM, menu petunjuk aplikasi dan melihat tentang Aplikasi.

2. *Class Diagram* : *Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-

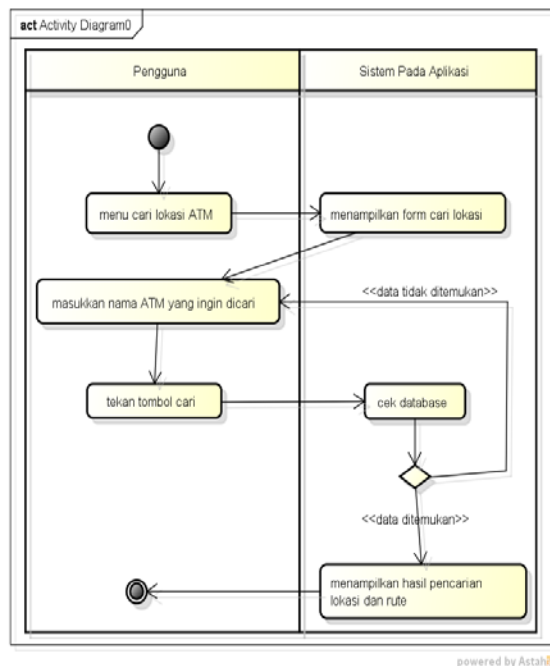
fungsi yang dimiliki oleh suatu kelas. *Class* diagram dari aplikasi yang akan dibangun dapat dilihat pada gambar 6.



Gambar 6. *Class Diagram* Aplikasi

Pada gambar di atas menjelaskan kelas-kelas yang terdapat pada sistem, terlihat bahwa ada tiga *class*, yaitu *class* poi, *class* atm, dan *class* route. Kelas tersebut terdiri dari tiga baris, baris pertama merupakan nama kelas, baris kedua merupakan *atribute* dan baris ketiga merupakan metode. Setiap kelas memiliki *atribute* dan metode yang berbeda-beda. Selain itu kelas-kelas tersebut juga saling berhubungan. Hubungan antar kelas yaitu *one-to-many*, contohnya pada kelas poi dengan kelas route, yang artinya satu poi memiliki banyak route. Hubungan *many-to-many*, contohnya kelas poi dengan atm yang artinya banyak poi memiliki banyak atm.

3. *Activity Diagram* : *Activity diagram* menyediakan analisis dengan kemampuan untuk memodelkan proses-proses dalam sistem informasi. *Activity diagram* dapat digunakan untuk memodelkan aliran kerja, *use case* individu atau logika keputusan yang didalamnya memuat metode individu. *Activity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. *Activity diagram* dari menu pencarian lokasi ATM dapat dilihat pada gambar 7.

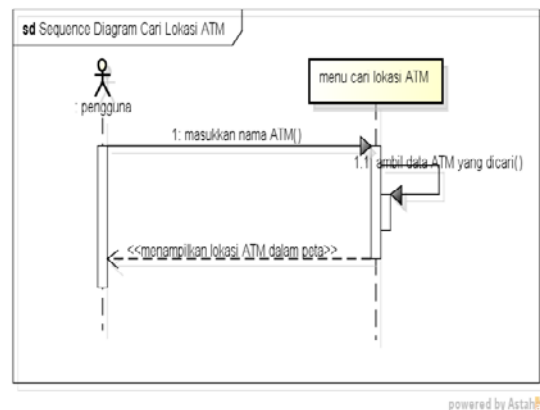


Gambar 7. Activity Diagram Menu Cari Lokasi ATM

Pada Gambar 7 merupakan aktifitas untuk menu cari lokasi ATM. Aktifitas ini diawali dengan pengguna memilih menu cari lokasi yang terdapat pada aplikasi. Setelah pengguna memilih menu tersebut, maka akan tampil *form* (halaman) cari lokasi. Pada halaman ini, pengguna memasukkan nama ATM yang akan dicari sebagai lokasi yang akan dituju. Setelah memasukkan nama ATM, sistem akan memeriksa ketersediaan data dalam *database*. Jika data ditemukan, maka sistem akan menampilkan informasi hasil pencarian lokasi yang berisi informasi tempat ATM, peta lokasi, dan rute yang ditempuh. Jika data tidak ditemukan, maka sistem akan menampilkan pesan *error* dan pengguna kembali memasukkan nama ATM dengan benar.

4. *Sequence Diagram* : Suatu *sequence diagram* adalah suatu penyajian perilaku yang tersusun sebagai rangkaian langkah-langkah percontohan dari waktu ke waktu. *Sequence diagram* digunakan untuk menggambarkan arus

pekerjaan, pesan yang disampaikan dan bagaimana elemen-elemen di dalamnya bekerja sama dari waktu ke waktu untuk mencapai suatu hasil. *Sequence diagram* dari menu pencarian lokasi ATM dapat dilihat pada gambar 8. Pengguna (*user*) memilih menu cari lokasi ATM, kemudian sistem akan menampilkan halaman pencarian. pengguna memasukkan nama ATM sebagai kata kunci pencarian. Selanjutnya sistem akan mencocokkan kata kunci yang dimasukkan dengan data yang tersimpan. Setelah kata kunci cocok, sistem akan mengambil data lokasi ATM tersebut dengan mengambil data titik koordinat berupa latitude dan longitude lokasi. Sebagai balasan dari interaksi yang dilakukan pengguna, sistem akan menampilkan informasi kepada user berupa hasil pencarian, peta lokasi, dan informasi tempat ATM yang dicari.



Gambar 8 Sequence Diagram Cari Lokasi ATM

V. HASIL DAN PEMBAHASAN

A. Implementasi Antar Muka

Adapun tampilan dan potongan *source code* untuk setiap menu pada aplikasi ini adalah sebagai berikut:

1. *Halaman Utama Aplikasi* : Tampilan halaman utama aplikasi dapat dilihat pada gambar 9.



Gambar 9. Tampilan Halaman Utama Aplikasi

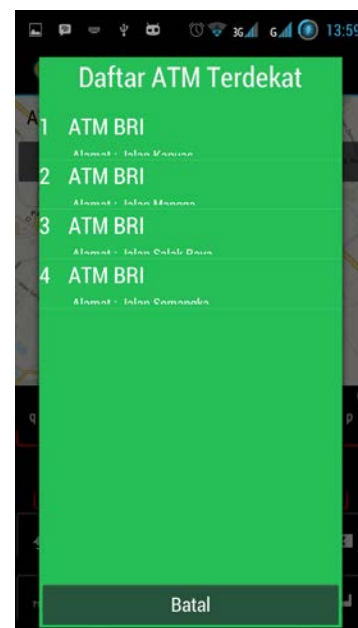
Halaman Utama merupakan halaman pertama yang akan tampil saat aplikasi dijalankan. Pada halaman ini terdapat judul aplikasi yaitu Pencarian ATM Terdekat di Kota Bengkulu dengan Metode *Kruskal*. Pada halaman utama ini terdapat tiga *button* menu yang dapat dipilih oleh pengguna yaitu menu Cari ATM Terdekat, menu Bantuan dan menu Tentang.

2. *Halaman Menu Cari Lokasi ATM* : Menu cari lokasi ATM merupakan menu yang menampilkan lokasi ATM dari posisi pengguna. Pada menu ini pengguna dapat mencari ATM terdekat dengan memasukkan nama ATM bank yang ingin dicari, kemudian sistem akan memberikan daftar ATM yang terdekat dari posisi pengguna. Pengguna juga dapat melihat informasi jarak lokasi ATM dengan pengguna serta informasi alamat lokasi ATM tersebut. Tampilan menu tambal ban terdekat dapat dilihat pada Gambar 5.2.



Gambar 10. Tampilan Halaman Menu Cari Lokasi ATM

Pada Gambar 10. terlihat tampilan halaman menu cari lokasi ATM terdapat *text box* untuk tempat memasukkan nama ATM yang akan dicari, kemudian setelah nama ATM dituliskan maka pilih *button* Tampilkan ATM Terdekat. Selanjutnya aplikasi akan menampilkan daftar lokasi ATM terdekat dengan nama yang telah dimasukkan pada *text box*.



Gambar 11. Tampilan Hasil Pencarian ATM Terdekat

3. *Halaman Menu Bantuan* : Menu bantuan merupakan menu yang berfungsi untuk membantu pengguna dalam mengoperasikan aplikasi ini. Pada menu bantuan ini berisikan penjelasan secara singkat cara penggunaan aplikasi. Tampilan menu bantuan aplikasi dapat dilihat pada Gambar 12 berikut :



Gambar 12. Tampilan Menu Bantuan Aplikasi

4. *Halaman Menu Tentang* : Menu tentang aplikasi merupakan menu yang menjelaskan tentang aplikasi yang dibuat. Tampilan menu tentang aplikasi dapat dilihat pada Gambar 13.



Gambar 13 Tampilan Menu Tentang Aplikasi

B. *Pengujian Metode Kruskal*

Aplikasi pencarian ATM ini dapat melakukan pencarian lokasi ATM terdekat dengan posisi pengguna. Proses pencarian untuk ATM terdekat pada aplikasi ini menggunakan algoritma Kruskal. Pengujian metode ini akan membandingkan hasil keluaran sistem untuk ATM terdekat terhadap penyelesaian secara manualnya.

Skenario pengujian dalam pengujian metode kruskal ini dilakukan pada 2 titik lokasi yang berbeda. Titik-titik tersebut berada di :

1. Alamat : Gang Van Iskandar Baksir VI
Pasar Jitra
Latitude : -3.794518
Longitude : 102.255588
Pencarian ATM : Bank Bengkulu, Bank BTN dan Bank Muamalat
2. Alamat : Jl. Mangga VD
Latitude : -3.82595
Longitude : 102.30596
Pencarian ATM : Bank Mandiri, Bank BTN dan Bank BCA.

Hasil untuk lokasi ATM terdekat dengan pengguna yang ditampilkan adalah titik lokasi ATM yang terhubung dengan titik lokasi pengguna setelah proses algoritma kruskal dijalankan.

Hasil pengujian metode Kruskal dapat dilihat pada Tabel 3 berikut:

Tabel 3 Hasil Pengujian Metode Kruskal

No	Titik Lokasi Pengguna	ATM Terdekat Yang Dicari	Jumlah ATM	Hasil
1.	-3.794518 / 102.255588	ATM Bank Bengkulu	9	Sukses
		ATM BTN	7	Sukses
		ATM Muamalat	5	Sukses
2.	-3.82595 / 102.30596	ATM Mandiri	13	Sukses
		ATM BTN	7	Sukses
		ATM BCA	9	Sukses

Dari pengujian dengan 2 lokasi yang berbeda didapatkan hasil seperti tabel 5.1 diatas. Dengan lokasi pengujian pertama yang berlokasi di Gang

Van Iskandar Baksir VI Pasar Jitra yang mencari 3 ATM berbeda yaitu ATM Bank Bengkulu yang mempunyai titik lokasi ATM 9 berhasil mendapatkan lokasi ATM terdekat sesuai dengan algoritma Kruskal yang digunakan, ATM BTN yang mempunyai titik lokasi ATM 7 berhasil mendapatkan lokasi ATM terdekat sesuai dengan algoritma Kruskal, dan ATM Muamalat yang mempunyai titik lokasi ATM 5 juga berhasil mendapatkan lokasi ATM terdekat sesuai dengan algoritma Kruskal. Kemudian pada lokasi pengujian kedua yang beralamat di Jl. Mangga VD No.02 dilakukan pencarian 3 ATM yang berbeda, yaitu ATM Mandiri yang mempunyai titik lokasi ATM sebanyak 13 berhasil mendapatkan lokasi ATM terdekat sesuai dengan algoritma Kruskal, ATM BTN yang mempunyai titik lokasi ATM 7 berhasil mendapatkan lokasi ATM terdekat sesuai dengan algoritma Kruskal, dan ATM BCA yang mempunyai titik lokasi ATM 10 juga berhasil mendapatkan lokasi ATM yang terdekat sesuai dengan algoritma Kruskal.

Sehingga sistem yang telah dibuat ini telah berhasil mengimplementasikan algoritma Kruskal untuk mencari lokasi ATM terdekat. Keberhasilan pengujian ini adalah 100% dari 6 kali pengujian.

C. Pengujian Keakuratan Jarak

Pada saat pencarian ATM terdekat sistem akan menampilkan informasi jarak dari posisi pengguna saat itu menuju posisi ATM terdekat. Jarak yang ditampilkan sistem perlu diuji keakuratannya. Pengujian dilakukan dengan membandingkan jarak yang dihasilkan sistem dengan jarak sesungguhnya. Perhitungan jarak yang ditampilkan pada sistem dihasilkan dari pengolahan data titik koordinat posisi pengguna dan titik koordinat posisi ATM menggunakan algoritma *Kruskal*. Sedangkan jarak sesungguhnya didapat dari perhitungan jarak posisi pengguna dengan posisi ATM menggunakan alat bantu hitung spidometer pada motor. Hasil perbandingan jarak yang dihasilkan sistem dengan jarak sesungguhnya pada hasil dapat dilihat pada tabel 4.

Pada tabel 4 terjadi perbedaan selisih jarak antara jarak yang dihasilkan sistem pencarian ATM terdekat dibandingkan dengan jarak sesungguhnya yang diuji dengan spidometer motor. Selisih jarak maksimum yang ditemui pada eksperimen adalah sebesar 10 m lebih dekat dibandingkan dengan jarak sesungguhnya dilapangan. Selisih jarak maksimum yang ditemui pada eksperimen adalah sebesar 10 m lebih dekat dibandingkan dengan jarak sesungguhnya dilapangan.

Tabel 4 Perbandingan Jarak Sistem Terhadap Jarak Sesungguhnya

No.	Lokasi Pengguna	Nama ATM	Alamat ATM	Jarak Pada Sistem (m)	Jarak Sesungguhnya (m)	Selisih (m)
1.	Gang Van Iskandar Baksir VI Pasar Jitra	ATM Bank Bengkulu	Jl. Sudirman	574	572	2
2.	Gang Van Iskandar Baksir VI Pasar Jitra	ATM BTN	Jl. Soeprapto	874	864	10
3.	Gang Van Iskandar Baksir VI Pasar Jitra	ATM Muamalat	Jl. S. Parman	1723	1726	3
4.	Jl. Mangga VD No.02	ATM Mandiri	Jl. Adam Malik	439	439	0
5	Jl. Mangga VD No.02	ATM Mandiri	Jl. Asahan	1537	1542	5
6.	Jl. Mangga VD No.02	ATM BTN	Jl. Adam Malik	441	441	0
7.	Jl. Mangga VD No.02	ATM BCA	Jl. Adam Malik	350	347	3
8	Jl. Mangga VD No.02	ATM BCA	Jl. Salak Raya	872	870	2

Perhitungan triangulasi membutuhkan minimal dua pemancar sinyal internet pada ponsel untuk dapat menghitung lokasi secara akurat, jika terdapat tiga atau lebih pemancar sinyal yang mendeteksi lokasi maka posisi lokasi tersebut dapat lebih akurat. Jika ponsel pengguna dalam keadaan sinyal yang lemah atau jauh dari pemancar maka kemungkinan akan menghasilkan perhitungan triangulasi yang kurang tepat. Hal inilah yang menyebabkan terjadinya perbedaan selisih antara jarak yang dihasilkan sistem dengan jarak sesungguhnya.

VI. KESIMPULAN

Berdasarkan analisa perancangan sistem, implementasi dan pengujian sistem, maka dapat disimpulkan bahwa:

1. Penelitian ini telah berhasil menghasilkan aplikasi pencarian lokasi ATM terdekat di Kota Bengkulu berbasis android dengan memanfaatkan fasilitas OpenStreetMap.
2. Berdasarkan uji coba yang dilakukan, algoritma *Kruskal* dapat mencari lokasi ATM terdekat berdasarkan lokasi pengguna.

VII. SARAN

1. Seharusnya perlu dilakukan pengembangan aplikasi yang mengizinkan pengguna untuk melakukan penambahan data, agar data lokasi ATM dapat terus diperbarui.
2. Sebaiknya sinyal ponsel pengguna ketika menggunakan aplikasi ini dalam kondisi sinyal kuat, karena sinyal lemah akan mengakibatkan pergeseran deteksi posisi pengguna.

REFERENSI

- [1] S Adiningsih. (2007)
[Online]. <http://st286324.sitekno.com/article/6528/persai-ngan-pada-industri-telepon-selular-diindonesia.html>
- [2] Atri Deliyah, *Aplikasi Peta Praktik Dokter Keluarga*

Badan Penyelenggara Jaminan Sosial Kesehatan (BPJS-Kesehatan). Bengkulu: Teknik Informatika UNIB, 2015.

- [3] I., Achmaliadi, R., Hanafi, I., Safitri, H., Kurniawan, I., & Pramono, A.H Natalia, *Dalam Geografi dan Koordinat Peta*. Bandung: Garis Pergerakan, 2005.
- [4] Riska, *Perbandingan Algoritma Kruskal Dengan Algoritma Genetika Dalam Penyelesaian Masalah Minimum Spanning Tree (MST)*. Bengkulu: Teknik Informatika UNIB, 2014.
- [5] Yustian Mantjoro. (2014) Penggunaan Teknologi Openstreetmap Untuk GIS Fasilitas Pelayanan Umum Berbasis Android (Studi Kasus Kota Palu). [Online]. <http://repo.pens.ac.id/1549/1/Makalah.pdf>
- [6] Fundation OpenStreetMap. (2011, May) OpenStreetMap. [Online]. <http://wiki.openstreetmap.org>